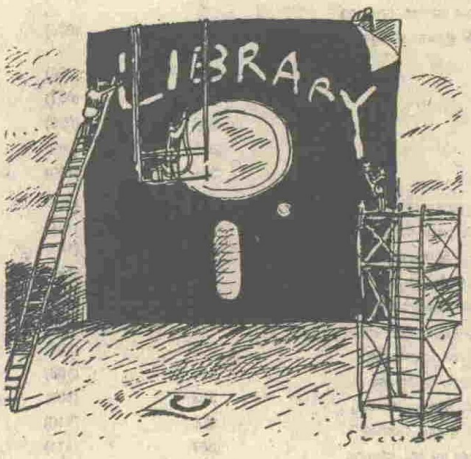
จเร เลิศสุดวิชัย

สร้างไลบรารี
เพิ่มเติมให้เทอร์โบซี

ภาษาซีมีคุณสมบัติพิเศษขยายขีดความสามารถได้ แต่เทอร์โบซีมีให้คุณมากกว่า



ภาพประกอบ : สุรชัย สหชาติโภคานันท์

ภาษาคอมพิวเตอร์ที่มีความสามารถในการเพิ่มเติมประสิทธิภาพตัวภาษาเองได้ เมื่อเปรียบเทียบกับภาษาอื่น เช่น โคบอล หรือ เบสิก ซึ่งมีลักษณะที่ไม่อ่อนตัว จะพบว่าความนิยมเริ่มลดน้อยลง ลักษณะของความสามารถที่เพิ่มขึ้นคือการยอมให้ผู้เขียนโปรแกรมสร้างสิ่งใหม่ ๆ ให้กับตัวภาษาได้ เช่น ในภาษาซีผู้เขียนโปรแกรมสามารถกำหนด (Declare) โครงสร้างข้อมูลแบบใหม่ได้สร้างและขยายฟังก์ชันสำหรับจัดการอินพุต/เอาต์พุตของระบบได้ นอกจากนั้นยังสามารถรวบรวมเอาฟังก์ชันต่าง ๆ เหล่านี้เก็บลงในไลบรารี่สำหรับใช้งานในภายหลังได้อีก ภาษาซีมีคุณสมบัติที่ขยายขีดความสามารถ เทอร์โบซีก็ยังมีคุณสมบัติการขยายมากกว่า เนื่องจากเทอร์โบซีถูกออกแบบสำหรับใช้งานบนเครื่องไอบีเอ็ม พีซี โดยเฉพาะ ดังนั้นตัวของเทอร์โบซีจึงมีคุณสมบัติเหมาะสำหรับการพัฒนาซอฟต์แวร์ต่าง ๆ บ
เครื่องไอบีเอ็ม พีซี สำหรับมุมภาษา
ในฉบับนี้และฉบับต่อไป จะเป็นเรื่อ
ของการสร้างฟังก์ชัน รวมทั้งจะเสน
เทคนิคและวิธีการใช้งานเทอร์โบซีใ
คุณสมบัติต่าง ๆ บางเรื่องที่เกี่ยวข้อ
กับฮาร์ดแวร์ เอ็มเอส-ดอส หรือ ไบออ
ผู้เขียนก็จะแทรกรายละเอียดที่จำเป็
อธิบายเสริมความเข้าใจต่างหาก เช่
บางฟังก์ชันอาจจำเป็นต้องเขียนด้ว
แอสเซมบลีในส่วนนั้นก็จะมีแอสเซมบ
แทรกลงไปในโปรแกรมภาษาซีเป็นอิน
ไลน์แอสเซมบลี (In-Line assembly
ดังนั้นนอกจากตัวโปรแกรมเทอร์โบซี
ต้องใช้แล้ว ยังต้องใช้เทอร์โบแอสเซม
เบลอร์ (TASM) ร่วมด้วยในบางครั้
อย่างไรก็ตาม การพัฒนาซอฟต์แวร์ด้ว
เทอร์โบซี ถ้าใช้ขีดความสามารถบา
อย่างแล้วก็จะทำให้โปรแกรมที่สร้างขึ้
ไม่มีคุณสมบัติในการโอนย้ายโปรแกร
(NON-portable) เช่นไม่สามารถน
โปรแกรมนั้นมาคอมพายล์ด้วยไมโคร
ซอฟต์ซีใหม่ได้ ไม่สามารถนำโปรแกร
ไปใช้ในงานใหม่บนเครื่องที่มีคอมพาย
เลอร์ตัวอื่น ๆ ได้

สำหรับในฉบับนี้จะเป็นการกล่า
ถึงคุณสมบัติพิเศษที่มีในเทอร์โบซีซึ่
จะถูกนำไปใช้ในการสร้างฟังก์ชันอื่น ๆ
ส่วนบทความตอนต่อไป อย่างไรก็ตา
มีหลายเรื่องที่ท่านผู้อ่านควรจะมีความรู้
มาก่อน หรือเตรียมตัวเพื่อเป็นพื้นฐาน
ในการที่จะเข้าใจคุณสมบัติพิเศษของ

อร์โบซี ซึ่งมีหัวข้อที่ต้องศึกษาก่อน นี้

– โครงสร้างภายในบางส่วนของ ไมโครโปรเซสเซอร์ 8088/8086/80286

– ฟังก์ชันสำหรับเรียกใช้งานบริการ องดอส

– ฟังก์ชันสำหรับเรียกใช้งานบริการ องไบออส

ะสำหรับคุณสมบัติพิเศษของเทอร์โบซี ะกล่าวก็มีดังนี้

– ตัวแปรรีจิสเตอร์ปลอม (Register eudo variable)

– อินไลน์ แอสเซมบลีในเทอร์โบซี

– ฟังก์ชันสำหรับอินเตอร์รัพต์

– ฟังก์ชันสำหรับการแสดงผลข้อ าม

– ฟังก์ชันแฝงในเทอร์โบซี

้าสู่เทอร์โบซี

เทอร์โบซีเป็นชุดคอมพายเลอร์ที่ อนข้างพิเศษคือนอกจากจะเป็นไปตาม าตรฐาน (ANSI) ของคอมพายเลอร์ ษาซีแล้วในเทอร์โบซียังมีเครื่องมือที่ รียมไว้ให้ผู้ใช้นำไปขยายขีดความ ามารถได้อีกมาก

เมื่อพูดถึงตำราภาษาซีต้องยกให้ บหนังสือ "The C Programming anguage" ของ Kernighan และ itchie ซึ่งถูกถือว่าเป็นคัมภีร์แห่ง ษาซีจนในที่สุดก็กลายเป็นมาตรฐาน ป เรื่องราวที่กล่าวถึงในหนังสือเล่มนี้ เป็นเรื่องทั่วไปในการเขียนโปรแกรม ษาซีแต่ในการพัฒนาโปรแกรมจริง ๆ ล้วงานส่วนใหญ่ของโปรแกรมก็จะมี วนที่เกี่ยวข้องกับการตอบสนองการ ดแป้นพิมพ์การแสดงผลทางจอภาพ งในภาษาซีเองก็มีฟังก์ชันมาตรฐาน ป็นไลบรารีให้ใช้งานแต่ฟังก์ชันเหล่า นั้นไม่ค่อยน่าใช้เท่าไร เนื่องจากยังคง ลักษณะเดิมของภาษาซีที่เป็นต้นฉบับ เอาไว้ ซึ่งอุปกรณ์อินพุต/เอาต์พุตใน สมัยก่อนยังล้าสมัยอยู่มาก ดังนั้นใน การทำงานของโปรแกรมส่วนการติดต่อ ผู้ใช้งาน (User Interface) จึงนับเป็น ส่วนสำคัญส่วนหนึ่ง ในตัวภาษาซีเองก็ นับได้ว่าเป็นภาษาที่อยู่ใกล้ชิดฮาร์ดแวร์ มากที่สุดภาษาหนึ่งและโดยเฉพาะเทอร์โบ-ซีซึ่งได้จัดเตรียมเครื่องมือเอาไว้เป็น พิเศษอยู่แล้ว การใช้งานต่าง ๆ กับ ฮาร์ดแวร์ของระบบพีซี ไบออสและ ดอสจะช่วยให้ประสิทธิภาพของโปร-แกรมดีขึ้นรวมทั้งระบบติดต่อผู้ใช้งาน จะน่าใช้งานมากขึ้นด้วย ฟังก์ชันหรือ ไลบรารีชุดที่จะพัฒนากันนับต่อไปนี้ก็ จะเกี่ยวข้องกับฮาร์ดแวร์ ไบออสและ ดอสโดยตรงในบางส่วน หรืออาจจะ นำภาษาแอสเซมบลีทั้งหมดมาใส่แทรก ไว้ในส่วนของอินไลน์แอสเซมบลี ก็ได้ คุณสมบัติที่โดดเด่นของเทอร์โบซีอันนี้ จะทำให้โปรแกรมที่พัฒนาสามารถใช้ งานฮาร์ดแวร์ได้อย่างใกล้ชิดมากที่สุด

ตัวแปรรีจิสเตอร์เทียม

ในการทำงานบางครั้งมีความ จำเป็นที่จะต้องใช้งานรีจิสเตอร์ของ ซีพียูโดยตรงซึ่งก็สามารถทำได้โดยอาศัย แอสเซมบลี ในเทอร์โบซีได้เตรียมตัว แปรพิเศษที่ใช้แทนรีจิสเตอร์ของซีพียู เพื่อให้โปรแกรมสามารถอ่านหรือเขียน ข้อมูลลงบนรีจิสเตอร์ไปซีพียูได้โดยผ่าน ตัวแปรเทียม ซึ่งจะมีชื่อเหมือนกับชื่อ ของรีจิสเตอร์เพียงมีตัวอักษรพิเศษ "__" อยู่หน้าชื่อตัวแปร ตัวอย่าง เช่น __AX, __AH, __AL, __DS, __FLAGE เป็นต้น รีจิสเตอร์เทียม สามารถใช้งานได้เหมือนตัวแปรทั่วไป คือสามารถกำหนดค่าตัวแปรให้กับตัวอื่น และใช้ในนิพจน์คณิตศาสตร์ได้ เนื่อง จากในตัวเทอร์โบซีเองใช้งานรีจิสเตอร์ หลายตัว ในการแปลภาษาซีให้เป็นภาษา เครื่อง เทอร์โบซีจำเป็นต้องใช้รีจิสเตอร์ บางตัวเพื่อให้งานเป็นไปตามวัตถุประสงค์ ดังนั้นจะพบว่าในบางโปรแกรมการใช้ งานตัวแปรรีจิสเตอร์เทียมที่ไม่ถูกวิธีจะ ทำให้โปรแกรมทำงานผิดพลาดได้ เช่น การกำหนดค่า AL ให้กับตำแหน่ง หน่วยความจำตำแหน่งหนึ่งโปรแกรม คอมพายเลอร์ภาษาซีอาจใช้รีจิสเตอร์ AL ตัวเดียวกันในการแทนตำแหน่ง หน่วยความจำตำแหน่งนั้นซึ่งผลของ การทำงานดังกล่าวก็จะผิดพลาดไป เพราะตัวเทอร์โบซีเองไม่สามารถจะ ทราบการใช้งานตัวแปรเทียมของผู้เขียน โปรแกรมได้ ดังนั้นการใช้งานตัวแปร เทียมผู้เขียนโปรแกรมจะต้องทราบเอง ว่าใช้รีจิสเตอร์เหล่านั้นทำอะไรอยู่

ในโปรแกรมรูปที่ 1 คือโปรแกรม ชื่อ Pseudo.C เป็นตัวอย่างการใช้งาน ตัวแปรรีจิสเตอร์เทียมในการทำความ

```
/*------------ pseudo.c ------------*/
/*
 * demostrates the use of register pseudo-variables
 */

#include <stdio.h>

int int_function (void);
long long_fuction (void);

void main ()
{
     int i;
     long j;

     i = int_function ();
     j = long_function ();
     printf ("\nthe int is: %d",  i);
     printf ("\nthe long is %ld", j);
}

int int_function (void)
{
     _AX = 1234;
}

long long_function (void)
{
     long lint = 654321L;

     _DX = (unsigned) (lint >> 16);
     _AX = (unsigned) (lint & 0xffff);
}
```

รูปที่ 1

เข้าใจผู้อ่านต้องทราบก่อนว่าในฟังก์ชันของเทอร์โบซีนั้น การส่งค่ากลับในกรณีของจำนวนเต็ม จะถูกส่งทางรีจิสเตอร์ AX สำหรับจำนวนเต็มแบบยาว (Long Integer) จะผ่านค่ากลับทางรีจิสเตอร์ DX กับ AX ฟังก์ชันหลักจะทำการเรียก ไอเอ็นที ฟังก์ชัน (int_function) ซึ่งส่งค่าจำนวนเต็มและลองก์ฟังก์ชัน (long_function) จะส่งค่าจำนวนเต็มแบบยาวออกมาว่า ภายในฟังก์ชันทั้งสองนั้นได้ทำการกำหนดค่าที่จะส่งกลับให้กับตัวแปรเทียม __AX และ __DX แทนการใช้คำสั่งรีเทอร์น อย่างไรก็ตามการคอมพายล์โปรแกรมนี้จะปรากฏข้อความเตือนให้ทราบว่าฟังก์ชันที่ประกาศไว้ไม่มีการส่งค่าเนื่องจากไม่พบคำสั่งรีเทอร์นภายในตัวฟังก์ชันซึ่งไม่จำเป็นต้องสนใจในคำเตือนเหล่านั้น

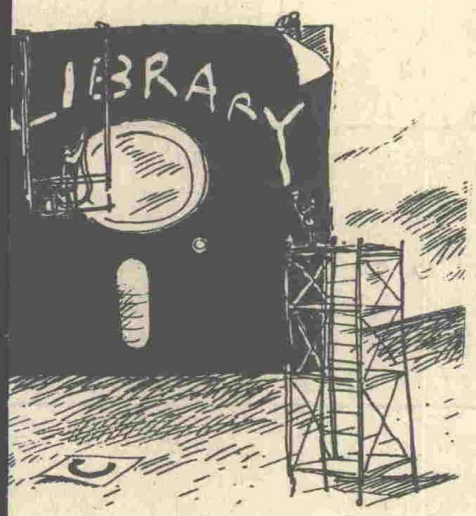
สรุปแล้วถ้าผู้อ่านตัดสินใจจะใช้ตัวแปรรีจิสเตอร์เทียมในโปรแกรมผู้อ่านจะต้องทราบวิธีการแปลภาษาของเทอร์โบซีว่ามีการใช้รีจิสเตอร์ชุดใดบ้างซึ่งตามปกติจะถูกกำหนดตายตัวไว้ รวมทั้งวิธีการส่งค่าเป็นพารามิเตอร์ระหว่างฟังก์ชัน (รายละเอียดมีอยู่ในคู่มือของเทอร์โบซี Turbo C User's Guide) อย่างไรก็ตามการใช้งานรีจิสเตอร์ในเทอร์โบซีแต่ละรุ่นจะแตกต่างกันไป หนทางที่จะช่วยเพิ่มความมั่นใจขึ้นด้วยการใช้ TCC.EXE ทำการแปลภาษาซีให้เป็นแอสแซมบลีโดยใช้สวิตช์ -S แล้วพิจารณาจากโค้ดแอสเซมบลีแทนว่ามีการใช้รีจิสเตอร์ในฟังก์ชันอย่างบ้าง

In-Line Assembly

ในส่วนของอินไลน์แอสเซมบ
ที่เสริมในเทอร์โบซีนี้เปิดให้ผู้เขียนโ
ภาษาเทอร์โบซีร่วมกับโปรแกรมภาษา
โดยมีรูปแบบการใช้งานดังนี้

```
asm <assembly language statement
```

เมื่อทำการแปลภาษาตัวเทอร์โบซี
สร้างไฟล์ผลลัพธ์เป็นไฟล์ภาษาแอสเซ
บลี โดยการนำแอสเซมบลีที่อยู่หลังคำ
asm ไปแทรกในโปรแกรมแอสเซมบ
จากนั้นจึงเรียกเอาเทอร์โบแอสเซ
เบลอร์ขึ้นมาทำงานกับไฟล์ผลลัพธ์จน
ไฟล์ที่เป็นออปเจ็กต์ (obj) ที่ต้อง
แปลภาษาแอสเซมบลีได้เองจึงต้อง
เทอร์โบแอสเซมเบอล์แปลภาษาแอ
เซมบลี จะพบว่าตัวเทอร์โบซีที่เ
ระบบรวมเบ็ดเสร็จ (Turbo C I
tegrated Development Environme
ไม่สามารถแปลภาษาซีที่ใช้อินไล
แอสเซมบลีได้ดังนั้นการแปลจึงต้
ใช้เทอร์โบซีคอมพายเลอร์ที่เป็นคอ
มานด์ไลน์ (TCC.EXE) ในการแปลโป
แกรมนั้น ๆ อย่างไรก็ตามถ้าต้องการ
ไฟล์โปรแกรมดังกล่าวในการพัฒนาโป
แกรมบนเทอร์โบซีที่เป็นระบบเบ็ดเส
ก็อาจจะแก้ไขได้ โดยการคอมพา
โปรแกรมเทอร์โบซีที่มีอินไลน์แอสเซมบ
แยกต่างหากด้วย TCC.EXE จากนั้
จึงนำเอาไฟล์ออปเจ็กต์ (.obj) ที่ได้ไป
ไว้ในไฟล์โปรเจ็กต์ (Project) ความจ
แล้วเทอร์โบซีเองไม่ใช่ตัวแปลภาษา
ตัวแรกที่มีอินไลน์แอสเซมบลีในตั
แปลภาษาซีตัวอื่น ๆ ก็มีส่วนนี้อยู่ด้ว
แต่รูปแบบการใช้งานก็จะต่างกันออกไ
ไม่มีมาตรฐาน



ฟังก์ชันแฝงในเทอร์โบซี

ในเทอร์โบซีมีอยู่หลายฟังก์ชันที่ลักษณะเหมือนมาโคร (macro) คือจะถูกสร้างเป็นคำสั่งภาษาแอสเซมบลีที่สั้นและตรงไปตรงมาข้อดีของฟังก์ชันพวกนี้คือจะทำงานได้รวดเร็วเพราะไม่มีโอเวอร์เฮด อย่างไรก็ตามในตัวแปลภาษาซีตัวอื่นอาจจะไม่มีฟังก์ชันแฝงดังนั้นถ้าหากใช้ฟังก์ชันแฝงพวกนี้ก็จะไม่สามารถนำเอาโปรแกรมดังกล่าวไปใช้งานในตัวแปลภาษาซีตัวอื่นได้ (NON-portable)

ตัวอย่างฟังก์ชันแฝงในเทอร์โบซีได้แก่

```
geninterrupt
enable
disable
inport
outport
inportb
outportb
peek
poke
peekb
pokeb
```

ฟังก์ชันพวกนี้จะถูกใช้งานมากในไลบรารีที่เราจะพัฒนาขึ้น จึงขออธิบายรายละเอียดวิธีการใช้งานคร่าว ๆ ดังนี้

สำหรับฟังก์ชัน geninterrupt เป็นฟังก์ชันที่ใช้เรียกโปรแกรมอินเตอร์รัพต์ ฟังก์ชันนี้ต้องการหมายเลขอินเตอร์รัพต์เป็นพารามิเตอร์ ตัวอย่างเช่น

```
geninterrupt (0x21);
```

เมื่อถูกแปลเทอร์โบซีจะแทรกคำสั่งภาษาแอสแซมบลีเข้าไปแทนคือคำสั่ง

```
INT 21 H
```

และยังสามารถนำเอาตัวแปรรีจิสเตอร์เทียมมาใช้งานร่วมกันได้ด้วย ดังตัวอย่างการอ่านตัวอักษรที่ถูกกดจากแป้นพิมพ์ผ่านไบออสดังต่อไปนี้

```
_AH = 0;
geninterrupt (0x16);
kbchar = _AL;
```

สำหรับฟังก์ชันเอนเนเบิล (enable) และดิสเอเบิล (disable) จะถูกแทนที่ด้วยคำสั่งแอสเซมบลี STI และ CLI ซึ่งเป็นคำสั่งสำหรับเซตและเคลียร์แฟล็กที่เกี่ยวกับอินเตอร์รัพต์ของซีพียูในรีจิสเตอร์แฟล็ก ฟังก์ชันนี้จะสามารถหยุดและอนุญาตอินเตอร์รัพต์จากภายนอกซีพียูได้ เช่นอินเตอร์รัพต์เวลาจากแป้นพิมพ์ มักจะใช้คำสั่งทั้งสองในกรณีที่ต้องใช้งานตัวแปรรีจิสเตอร์เทียม _SS และ _SP (เกี่ยวกับรีจิสเตอร์สแต็กเซกเมนต์และรีจิสเตอร์สแต็กพอยเตอร์) เพราะว่าในระหว่างกำลังจะทำการเปลี่ยนแปลงค่าในรีจิสเตอร์ทั้งสอง ถ้ามีอินเตอร์รัพต์เกิดขึ้น อาจจะทำให้เกิดเหตุการณ์ที่ไม่พึงปรารถนาขึ้นได้ ฟังก์ชันอินพอร์ต (inport) และเอาต์-พอร์ต (outport) ก็เป็นฟังก์ชันแฝงอีกกลุ่ม (คู่กับฟังก์ชัน inportb และ outportb) ซึ่งเมื่อถูกแปลก็จะถูกแทนที่ด้วยคำสั่ง IN และ OUT ส่วนฟังก์ชันโพค (poke) และ พีค (peek) จะเป็นการอ้างอิงถึงตำแหน่งหน่วยความจำโดยใช้พอยน์เตอร์เข้าช่วย (เป็นมาโคร) แต่ฟังก์ชันตัวจริงของกลุ่มนี้เกี่ยวกับ port และฟังก์ชันโพค/พีคในไลบรารีของเทอร์โบซีดังนั้นในโปรแกรมที่ใช้คำสั่งเหล่านี้ถ้าไม่ได้ดึงไฟล์ DOS.H มาแปลด้วย (ใช้คำสั่ง # include <dos.H> เทอร์โบซีจะเรียกฟังก์ชันตัวจริงในไลบรารีมาใช้งานแทนฟังก์ชันแฝงกลุ่มนี้

ฟังก์ชันสำหรับอินเตอร์รัพต์

ในระบบคอมพิวเตอร์ที่มีอุปกรณ์หลาย ๆ ชนิด ซีพียูต้องคอยให้บริการแก่อุปกรณ์ที่ร้องขอมาการเรียกอินเตอร์รัพต์และการบริการให้แต่ละอุปกรณ์ก็ต่างกันออกไปซีพียูต้องบริการให้ถูกต้องตามอุปกรณ์เหล่านั้น ในระบบของไมโครคอมพิวเตอร์ระดับพีซีอินเตอร์รัพต์ จะมีหมายเลขประจำ และงานบริการของแต่ละอินเตอร์รัพต์จะถูกจัดเก็บอยู่หน่วยความจำแรกสุดของระบบที่เรียกว่า ตารางอินเตอร์รัพต์ (Interrupt Vector) เมื่อเกิดการอินเตอร์รัพต์ขึ้นซีพียูจะต้องให้บริการแก่อินเตอร์รัพต์จนเสร็จแล้วจึงกลับมาทำงานที่ค้างอยู่ ดังนั้นก่อนซีพียูจะเข้าไปทำงานบริการจึงจำเป็นต้องเก็บสถานะต่าง ๆ ของระบบเอาไว้เช่นเก็บค่าในรีจิสเตอร์ต่าง ๆ เอาไว้ในสแต็กและคืนเมื่อทำงานบริการเสร็จนอกจากรีจิสเตอร์ทั่วไปที่จะต้องเก็บแล้วรีจิสเตอร์แฟล็กเป็นอีกตัวหนึ่งที่สำคัญดังนั้นเมื่อเกิดอินเตอร์รีจิสเตอร์แฟล็กก็จะถูกเก็บลงไปในสแต็กด้วยเช่นเดียวกับรีจิสเตอร์ทั่วไป ในทำนองเดียวกันขณะออกจากบริการอินเตอร์รัพต์

ก็จะต้องดึงค่ารีจิสเตอร์แฟล็กคืนด้วย ซึ่งจะมีคำสั่งพิเศษคือ IRET คำสั่งจะดึงค่ารีจิสเตอร์คืนรวมทั้งรีจิสเตอร์แฟล็กด้วย โดยทั่วไปฟังก์ชันบริการอินเตอร์รัพต์มักจะถูกเขียนด้วยภาษาแอสเซมบลีหรืออย่างน้อยที่สุดก็ใช้อินไลน์แอสเซมบลีของคอมพายเลอร์ตัวนั้นแต่ในเทอร์โบซีมีคำพิเศษคืออินเตอร์รัพต์ ซึ่งถ้าชื่อนี้ถูกประกาศอยู่หน้าฟังก์ชันใดจะทำหน้าที่เป็นฟังก์ชันบริการอินเตอร์รัพต์เมื่อถูกเรียกใช้งานก็จะทำการเก็บ (push) รีจิสเตอร์ทั้งหมดลงสแต็ก กำหนดรีจิสเตอร์ DS ให้ชี้ไปยังดาต้าเซกเมนต์ของตัวโปรแกรมและกำหนดรีจิสเตอร์ BP เป็นค่าเดียวกับรีจิสเตอร์ SP ในขณะนั้น จะเห็นว่าทุกครั้งเมื่อมีการอินเตอร์รัพต์ฟังก์ชันบริการดังกล่าวจะถูกเรียกใช้และในขณะทำงานฟังก์ชันบริการจะใช้งานสแต็กบริเวณที่ก่อนจะถูกอินเตอร์รัพต์ซึ่งในคู่มือจะแนะนำว่าการคอมพายล์โปรแกรมควรจะเลือกให้ไม่มีการตรวจสอบสแต็กในการเก็บรีจิสเตอร์ทั้งหมดลงสแต็กและดึงคืนในฟังก์ชันที่ให้เป็นฟังก์ชันบริการอินเตอร์รัพต์นั้น จะเป็นเสมือนการส่งผ่านรีจิสเตอร์ทั้งชุดให้กับฟังก์ชันบริการทางสแต็ก ดังนั้นค่าในสแต็กบริการดังกล่าวสามารถถูกมองเป็นพารามิเตอร์ที่ถูกส่งผ่านให้ฟังก์ชันและถ้าหากมีการเปลี่ยนค่าพารามิเตอร์เมื่อจบการทำงานจะมีการดึงค่า (pop) คืนให้กับชุดรีจิสเตอร์และค่าในรีจิสเตอร์ก็จะถูกเปลี่ยนค่าไปด้วย ลักษณะนี้ทำให้สามารถพัฒนาฟังก์ชันบริการอินเตอร์รัพต์ที่ทำงานเหมือนฟังก์ชันบริการในไบออสและดอสได้นั่นเอง

ตัวอย่างการประกาศฟังก์ชันบริการอินเตอร์รัพต์

```
typedef
    {
    int bp, di, si, ds, es, dx, cx, bx, ax, ip, cs, fl ;
    } IREGS ;
void interrupt newisr (IREGS iregs)
```

รูปที่ 2

```
void interrupt newisr (bp, di, si,
ds, es, dx, cx, bx, ax, ip, cs, fl,
int bp, di, si  ds, es, dx, cx, bx,
ax, ip, cs, fl ;
```

หรืออาจประกาศเป็นโครงสร้างข้อมูลขึ้นมาใหม่เพื่อความสะดวกในการใช้งานก็ได้ดังรูปที่ 2

สังเกตจะเห็นลำดับการเรียงชื่อพารามิเตอร์เหมือนและตรงกันกับลำดับการเก็บค่าในสแต็ก

และเราสามารถจะทำการกำหนดค่าใหม่ให้กับพารามิเตอร์ตามตัวอย่างใหม่ได้โดยจะมีผลต่อรีจิสเตอร์ตัวนั้น ๆ ตามลำดับ เช่น

```
ax = 123;
หรือ
iregs.ax = 123;
```

รายละเอียดการใช้งานฟังก์ชันบริการอินเตอร์รัพต์จะกล่าวถึงอย่างละเอียดอีกครั้งในตอนต่อไป

ฟังก์ชันการแสดงผลข้อความ

ในเทอร์โบซีนอกจากฟังก์ชันที่เกี่ยวกับระบบอินพุต/เอาต์พุตที่เป็นไปตามมาตรฐาน ANSI X3J11 แล้ว ยังได้เพิ่มฟังก์ชันพิเศษสำหรับระบบเครื่องพีซีที่มีความหลากหลายในอุปกรณ์ เช่น การวัดควบคุมการแสดงผลตั้งแต่ CGA จนถึง VGA ฟังก์ชันการแสดงผลในเทอร์โบซีสามารถใช้งานได้กับการ์ดดังกล่าวโดยไม่ต้องดัดแปลงแก้ไขโปรแกรมนอกจากนั้นยังสามารถทำงานบนระบบวินโดว์ที่เทอร์โบซีเตรียมเอาไว้ให้ด้วย ซึ่งดูได้จาก "Turbo C Reference Guide" ในบทที่ 1 ที่เป็นเรื่องของ "Text Window Display Routine" ตัวอย่างฟังก์ชันได้แก่

```
cprintf
cputs
cscanf
gefche
putch
getpass
```

สรุป

ในฉบับนี้ก็จะเริ่มทำการพัฒนาฟังก์ชันเพิ่มขีดความสามารถให้แก่เทอร์โบซีแต่อย่างไรก็ตามท่านผู้อ่านควรทบทวนและทำความเข้าใจเพิ่มเติมในสิ่งที่กล่าวมาแล้วในฉบับนี้เพื่อเตรียมตัวไว้ในตอนต่อไป